

What is claimed is:

Sub
AI-1

1 1. A method for use in a database system, comprising:
2 storing a materialized join view based on at least two base
3 relations;
4 storing at least one auxiliary relation containing one or more
5 attributes of one of the base relations, the auxiliary relation partitioned according
6 to a join attribute; and
7 updating the at least one auxiliary relation in response to
8 modification of the one base relation.

1 2. The method of claim 1, further comprising storing the one base
2 relation that is not partitioned according to the join attribute.

1 3. The method of claim 2, further comprising:
2 receiving a tuple into the database system;
3 storing the tuple in one of the at least two base relations;
4 storing the tuple in the auxiliary relation; and
5 using the auxiliary relation to determine whether to update the
6 materialized join view.

1 4. A method comprising:
2 receiving a tuple into a relation at a first node, wherein the tuple
3 comprises a join attribute and the relation is not partitioned according to the join
4 attribute;
5 storing the tuple in an auxiliary relation at a second node, wherein
6 the auxiliary relation is partitioned according to the join attribute;
7 identifying second tuples of a second relation;
8 joining the tuple with the second tuples to produce join results; and
9 storing the join results in a join view.

09900280-070601
T09020-0820060

1 5. The method of claim 4, wherein joining the tuple with the second
2 tuples to produce join results comprises:
3 identifying second join attributes in the second tuples; and
4 comparing the second join attributes with the join attribute of the
5 relation.

1 6. The method of claim 5, wherein identifying second tuples of a
2 second relation comprises identifying second tuples of the second relation at the
3 second node.

1 7. The method of claim 4, wherein storing the tuple in an auxiliary
2 relation at a second node comprises:
3 determining that a join view definition excludes an attribute of the
4 tuple; and
5 not storing the excluded attribute in the auxiliary relation.

1 8. The method of claim 4, wherein storing the tuple in an auxiliary
2 relation at a second node comprises:
3 determining that a join view definition includes a condition on one
4 of the attributes of the tuple; and
5 identifying the attribute in the tuple;
6 determining that the condition is not met; and
7 not storing the tuple in the auxiliary relation.

1 9. The method of claim 4, wherein storing the tuple in an auxiliary
2 relation at a second node comprises:
3 determining that a join view definition includes a condition that
4 cannot be met; and
5 not storing the tuple in the auxiliary relation.

10. The method of claim 5, wherein storing the tuple in an auxiliary relation at a second node comprises:

- determining that the join attribute is a key of the relation;
- determining that the second join attribute is a foreign key of a second relation, wherein the foreign key references to the join attribute; and
- not maintaining a second auxiliary relation.

11. The method of claim 4, wherein storing the join results in a join view comprises:

- determining that a join view definition includes a condition on one of the attributes of the tuple; and
- identifying the attribute in the tuple;
- determining that the condition is not met; and
- not storing the join results in the join view.

12. The method of claim 4, wherein storing the join results in a join view comprises:

- determining that a join view definition includes a condition that cannot be met; and
- not storing the join results in the join view.

13. A database system comprising:

- a storage module to store relations and at least one auxiliary relation corresponding to one of the relations, the at least one auxiliary relation containing one or more attributes of the one relation, the at least one auxiliary relation partitioned differently than the one relation, the storage module further to store a join view based on a join of the relations; and

a controller adapted to update the join view using the at least one auxiliary relation.

1 14. The database system of claim 13, wherein the controller is further
2 adapted to receive a tuple and store the tuple in one of the relations and in one
3 of the at least one auxiliary relations.

1 15. The database system of claim 14, wherein the controller is further
2 adapted to not update the join view after receiving some tuples.

1 16. An article comprising a medium storing instructions for enabling a
2 processor-based system to:

3 receive a tuple into a relation at a first node, wherein the tuple
4 comprises a join attribute and the relation is not partitioned according to the join
5 attribute;

6 store the tuple in an auxiliary relation at a second node, wherein
7 the auxiliary relation is partitioned according to the join attribute;

8 identify second tuples of a second relation;

9 join the tuple with the second tuples to produce join results; and

10 store the join results in a join view.

1 17. The article of claim 16, further storing instructions for enabling the
2 processor-based system to:

3 identify second join attributes in the second tuples; and

4 compare the second join attributes with the join attribute of the
5 relation.

1 18. The article of claim 16, further storing instructions for enabling the
2 processor-based system to:

3 determine that a join view definition excludes an attribute of the
4 tuple; and

5 not store the excluded attribute in the auxiliary relation.

1 19. The article of claim 16, further storing instructions for enabling the
2 processor-based system to:

3 determine that a join view definition includes a condition on one of
4 the attributes of the tuple; and

5 identify the attribute in the tuple;

6 determine that the condition is not met; and

7 not store the tuple in the auxiliary relation.

1 20. The article of claim 16, further storing instructions for enabling the
2 processor-based system to:

3 determine that a join view definition includes a condition that
4 cannot be met; and

5 not store the tuple in the auxiliary relation.

1 21. The article of claim 17, further storing instructions for enabling the
2 processor-based system to:

3 determine that the join attribute is a key of the relation;

4 determine that the second join attribute is a foreign key of a
5 second relation, wherein the foreign key references to the join attribute; and

6 not maintain a second auxiliary relation.

1 22. The article of claim 16, further storing instructions for enabling the
2 processor-based system to:

3 determine that a join view definition includes a condition on one of
4 the attributes of the tuple; and

5 identify the attribute in the tuple;

6 determine that the condition is not met; and

7 not store the join results in the join view.

1 23. The article of claim 16, further storing instructions for enabling the
2 processor-based system to:

3 determine that a join view definition includes a condition that
4 cannot be met; and

5 not store the join results in the join view.

*add
A2*

109070" 08200660